

In [1]:

```
import pandas as pd
```

In [2]:

```
df = pd.read_csv('customer_acq.csv')
```

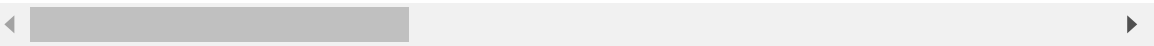
In [3]:

```
df.head()
```

Out[3]:

	food_category	food_department	food_family	store_sales(in millions)	store_cost(in millions)	unit_sales(in millions)	
0	Breakfast Foods	Frozen Foods	Food	7.36	2.7232	4	
1	Breakfast Foods	Frozen Foods	Food	5.52	2.5944	3	
2	Breakfast Foods	Frozen Foods	Food	3.68	1.3616	2	
3	Breakfast Foods	Frozen Foods	Food	3.68	1.1776	2	
4	Breakfast Foods	Frozen Foods	Food	4.08	1.4280	3	

5 rows × 40 columns



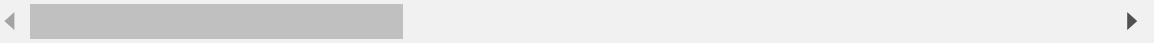
In [4]:

```
df.tail()
```

Out[4]:

	food_category	food_department	food_family	store_sales(in millions)	store_cost(in millions)	unit_sales(million
60423	Specialty	Carousel	Non-Consumable	2.76	1.3248	
60424	Specialty	Carousel	Non-Consumable	1.60	0.4960	
60425	Specialty	Carousel	Non-Consumable	5.52	2.5392	
60426	Specialty	Carousel	Non-Consumable	8.28	2.5668	
60427	Specialty	Carousel	Non-Consumable	9.20	4.2320	

5 rows × 40 columns



In [5]:

```
df.shape
```

Out[5]:

```
(60428, 40)
```

In [6]:

```
df.columns
```

Out[6]:

```
Index(['food_category', 'food_department', 'food_family',  
      'store_sales(in millions)', 'store_cost(in millions)',  
      'unit_sales(in millions)', 'promotion_name', 'sales_country',  
      'marital_status', 'gender', 'total_children', 'education',  
      'member_card', 'occupation', 'houseowner', 'avg_cars_at home(appro  
x)',  
      'avg. yearly_income', 'num_children_at_home',  
      'avg_cars_at home(approx).1', 'brand_name', 'SRP', 'gross_weight',  
      'net_weight', 'recyclable_package', 'low_fat', 'units_per_case',  
      'store_type', 'store_city', 'store_state', 'store_sqft', 'grocery_s  
qft',  
      'frozen_sqft', 'meat_sqft', 'coffee_bar', 'video_store', 'salad_ba  
n',  
      'prepared_food', 'florist', 'media_type', 'cost'],  
      dtype='object')
```

In [8]:

```
df.duplicated().sum()
```

Out[8]:

```
0
```

In [9]:

```
df.isnull().sum()
```

Out[9]:

```
food_category          0
food_department        0
food_family            0
store_sales(in millions)  0
store_cost(in millions)  0
unit_sales(in millions)  0
promotion_name         0
sales_country          0
marital_status         0
gender                 0
total_children         0
education              0
member_card           0
occupation             0
houseowner             0
avg_cars_at home(approx)  0
avg. yearly_income     0
num_children_at_home    0
avg_cars_at home(approx).1  0
brand_name             0
SRP                    0
gross_weight           0
net_weight             0
recyclable_package     0
low_fat                0
units_per_case         0
store_type             0
store_city             0
store_state            0
store_sqft             0
grocery_sqft           0
frozen_sqft            0
meat_sqft              0
coffee_bar            0
video_store            0
salad_bar              0
prepared_food          0
florist                0
media_type             0
cost                   0
dtype: int64
```

In [10]:

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60428 entries, 0 to 60427
Data columns (total 40 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   food_category                             60428 non-null  object
1   food_department                           60428 non-null  object
2   food_family                               60428 non-null  object
3   store_sales(in millions)                  60428 non-null  float64
4   store_cost(in millions)                   60428 non-null  float64
5   unit_sales(in millions)                   60428 non-null  int64
6   promotion_name                            60428 non-null  object
7   sales_country                             60428 non-null  object
8   marital_status                            60428 non-null  object
9   gender                                    60428 non-null  object
10  total_children                            60428 non-null  int64
11  education                                 60428 non-null  object
12  member_card                              60428 non-null  object
13  occupation                                60428 non-null  object
14  houseowner                               60428 non-null  object
15  avg_cars_at home(approx)                  60428 non-null  int64
16  avg. yearly_income                        60428 non-null  object
17  num_children_at_home                      60428 non-null  int64
18  avg_cars_at home(approx).1                60428 non-null  int64
19  brand_name                                60428 non-null  object
20  SRP                                        60428 non-null  float64
21  gross_weight                             60428 non-null  float64
22  net_weight                               60428 non-null  float64
23  recyclable_package                       60428 non-null  int64
24  low_fat                                  60428 non-null  int64
25  units_per_case                           60428 non-null  int64
26  store_type                               60428 non-null  object
27  store_city                               60428 non-null  object
28  store_state                              60428 non-null  object
29  store_sqft                               60428 non-null  int64
30  grocery_sqft                             60428 non-null  int64
31  frozen_sqft                              60428 non-null  int64
32  meat_sqft                                60428 non-null  int64
33  coffee_bar                               60428 non-null  int64
34  video_store                              60428 non-null  int64
35  salad_bar                                60428 non-null  int64
36  prepared_food                             60428 non-null  int64
37  florist                                  60428 non-null  int64
38  media_type                               60428 non-null  object
39  cost                                      60428 non-null  float64
dtypes: float64(6), int64(17), object(17)
memory usage: 18.4+ MB
```

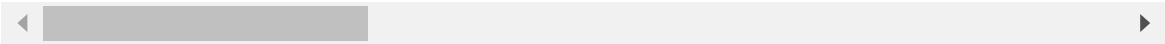
In [11]:

```
df.describe()
```

Out[11]:

	store_sales(in millions)	store_cost(in millions)	unit_sales(in millions)	total_children	avg_cars_at home(approx)	num_children
count	60428.000000	60428.000000	60428.000000	60428.000000	60428.000000	60428.000000
mean	6.541031	2.619460	3.093169	2.533875	2.200271	2.200271
std	3.463047	1.453009	0.827677	1.490165	1.109644	1.109644
min	0.510000	0.163200	1.000000	0.000000	0.000000	0.000000
25%	3.810000	1.500000	3.000000	1.000000	1.000000	1.000000
50%	5.940000	2.385600	3.000000	3.000000	2.000000	2.000000
75%	8.670000	3.484025	4.000000	4.000000	3.000000	3.000000
max	22.920000	9.726500	6.000000	5.000000	4.000000	4.000000

8 rows × 7 columns



In [12]:

```
df.nunique()
```

Out[12]:

```
food_category          45
food_department        22
food_family            3
store_sales(in millions) 1033
store_cost(in millions) 9919
unit_sales(in millions) 6
promotion_name         49
sales_country          3
marital_status         2
gender                 2
total_children         6
education              5
member_card            4
occupation             5
houseowner             2
avg_cars_at home(approx) 5
avg. yearly_income     8
num_children_at_home   6
avg_cars_at home(approx).1 5
brand_name            111
SRP                   315
gross_weight          376
net_weight            332
recyclable_package     2
low_fat               2
units_per_case        36
store_type             5
store_city            19
store_state           10
store_sqft            20
grocery_sqft          20
frozen_sqft           20
meat_sqft             20
coffee_bar            2
video_store           2
salad_bar             2
prepared_food          2
florist               2
media_type            13
cost                  328
dtype: int64
```

In [13]:

```
obj_cols = df.select_dtypes(include=['object']).columns
print('Object columns:', obj_cols)
```

```
Object columns: Index(['food_category', 'food_department', 'food_family',
                        'promotion_name',
                        'sales_country', 'marital_status', 'gender', 'education', 'member_c
ard',
                        'occupation', 'houseowner', 'avg. yearly_income', 'brand_name',
                        'store_type', 'store_city', 'store_state', 'media_type'],
                        dtype='object')
```

In [15]:

```
import numpy as np
```

In [16]:

```
num_cols = df.select_dtypes(include=np.number).columns
print('Numerical columns:', num_cols)
```

```
Numerical columns: Index(['store_sales(in millions)', 'store_cost(in millions)',
                          'unit_sales(in millions)', 'total_children', 'avg_cars_at home(approx)',
                          'num_children_at_home', 'avg_cars_at home(approx).1', 'SRP',
                          'gross_weight', 'net_weight', 'recyclable_package', 'low_fat',
                          'units_per_case', 'store_sqft', 'grocery_sqft', 'frozen_sqft',
                          'meat_sqft', 'coffee_bar', 'video_store', 'salad_bar', 'prepared_food',
                          'florist', 'cost'],
                          dtype='object')
```

In [17]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [18]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [21]:

```
for i in obj_cols:
    print(i)
    print(df[i].unique())
    print('\n')
```

```
food_category
['Breakfast Foods' 'Bread' 'Canned Shrimp' 'Baking Goods' 'Vegetables'
 'Frozen Desserts' 'Candy' 'Snack Foods' 'Dairy' 'Starchy Foods'
 'Cleaning Supplies' 'Decongestants' 'Meat' 'Hot Beverages'
 'Jams and Jellies' 'Carbonated Beverages' 'Seafood' 'Specialty'
 'Kitchen Products' 'Electrical' 'Beer and Wine' 'Candles' 'Fruit'
 'Pure Juice Beverages' 'Canned Soup' 'Paper Products' 'Canned Tuna'
 'Eggs' 'Hardware' 'Canned Sardines' 'Canned Clams' 'Pain Relievers'
 'Side Dishes' 'Bathroom Products' 'Magazines' 'Frozen Entrees' 'Pizza'
 'Cold Remedies' 'Canned Anchovies' 'Drinks' 'Hygiene' 'Plastic Product
s'
 'Canned Oysters' 'Packaged Vegetables' 'Miscellaneous']
```

```
food_department  
['Frozen Foods' 'Baked Goods' 'Canned Foods' 'Baking Goods' 'Produce'  
 'Snacks' 'Snack Foods' 'Dairy' 'Starchy Foods' 'Household'  
 'Health and Hygiene' 'Meat' 'Beverages' 'Seafood' 'Deli'  
 'Alcoholic Beverages' 'Canned Products' 'Eggs' 'Periodicals'  
 'Bulk Food' 'Food Labels' 'Specialty Foods']
```

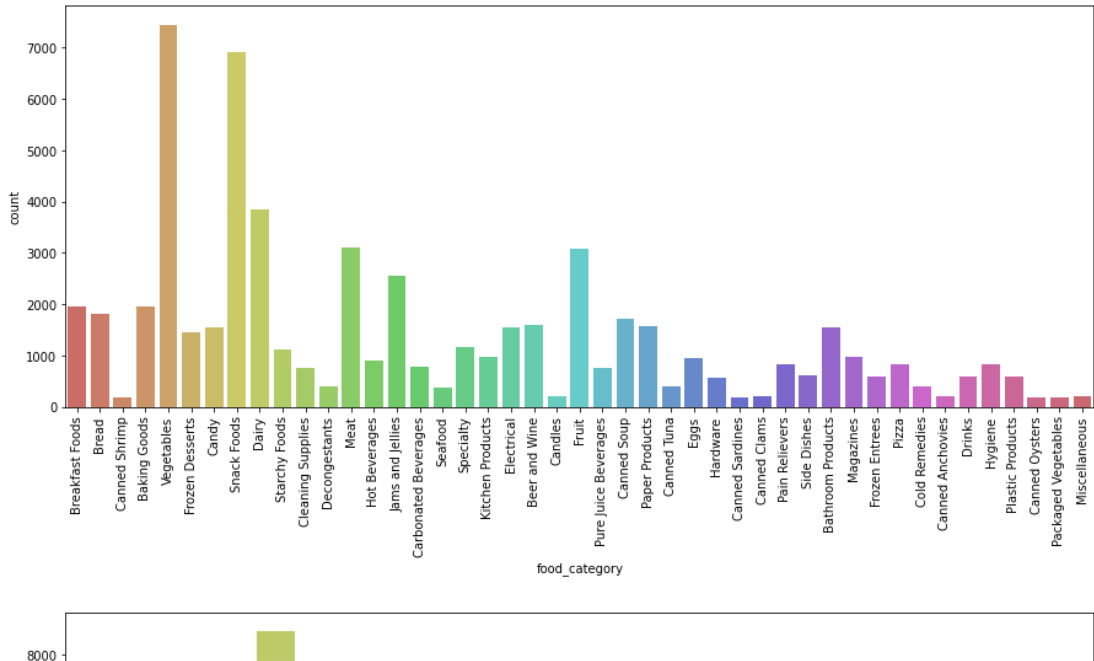
In [22]:

```
for i in obj_cols:
    print(i)
    print(df[i].value_counts())
    print('\n')
```

food_category	
Vegetables	7440
Snack Foods	6919
Dairy	3835
Meat	3107
Fruit	3080
Jams and Jellies	2550
Baking Goods	1947
Breakfast Foods	1946
Bread	1797
Canned Soup	1722
Beer and Wine	1590
Paper Products	1568
Bathroom Products	1552
Electrical	1544
Candy	1538
Frozen Desserts	1446
Specialty	1174
Starchy Foods	1103
..	...

In [24]:

```
for i in obj_cols:
    plt.figure(figsize=(15,6))
    sns.countplot(df[i], data = df, palette = 'hls')
    plt.xticks(rotation = 90)
    plt.show()
```

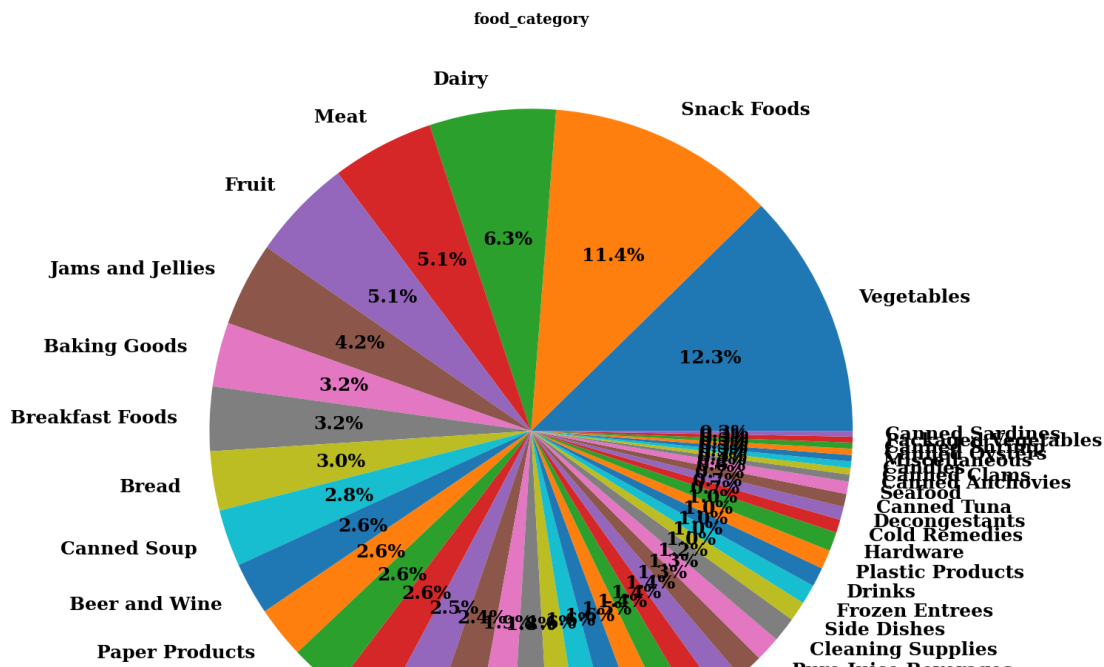


In [25]:

```

for i in obj_cols:
    plt.figure(figsize=(30,20))
    plt.pie(df[i].value_counts(), labels=df[i].value_counts().index, autopct='%1.1f%%',
            hfont = {'fontname':'serif', 'weight': 'bold'})
    plt.title(i, size=20, **hfont)
    plt.show()

```

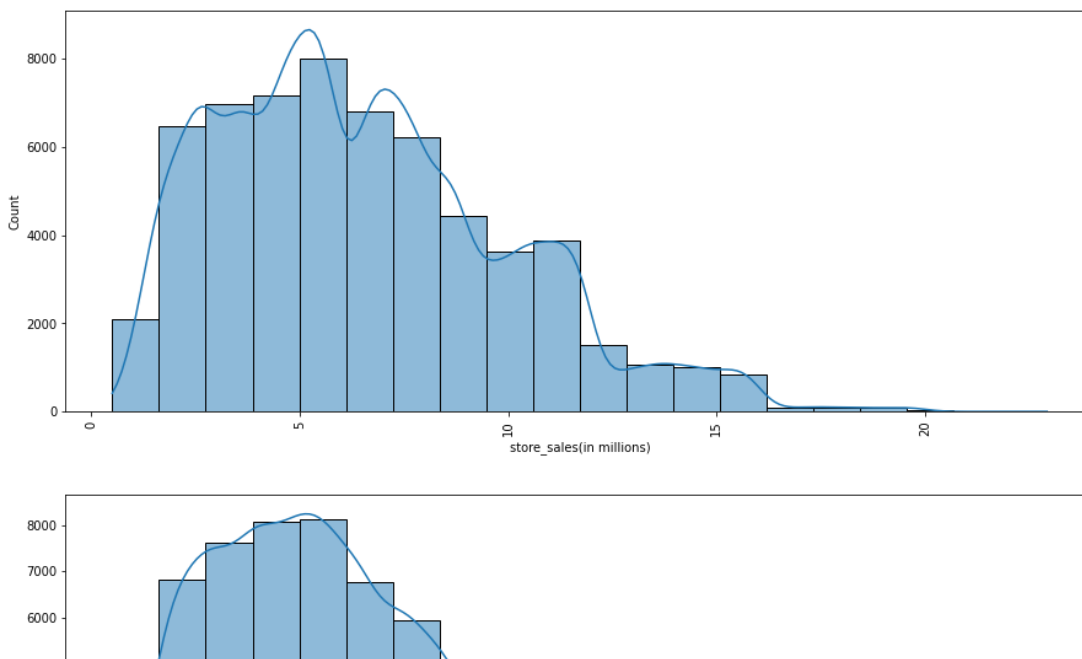


In [26]:

```

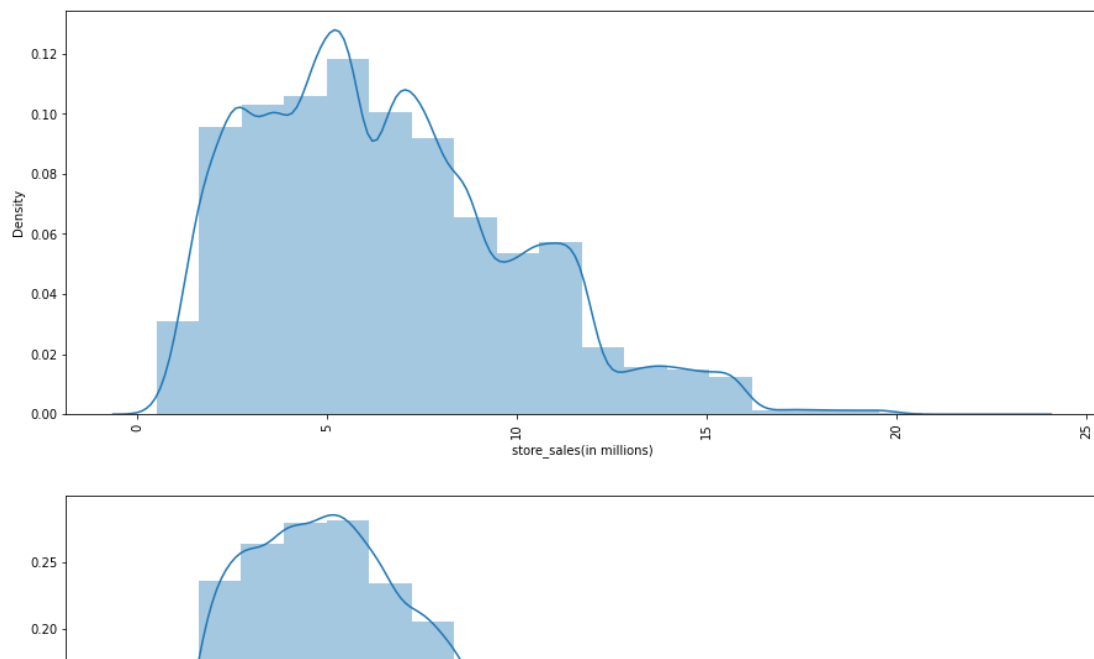
for i in num_cols:
    plt.figure(figsize=(15,6))
    sns.histplot(df[i], kde = True, bins = 20, palette = 'hls')
    plt.xticks(rotation = 90)
    plt.show()

```



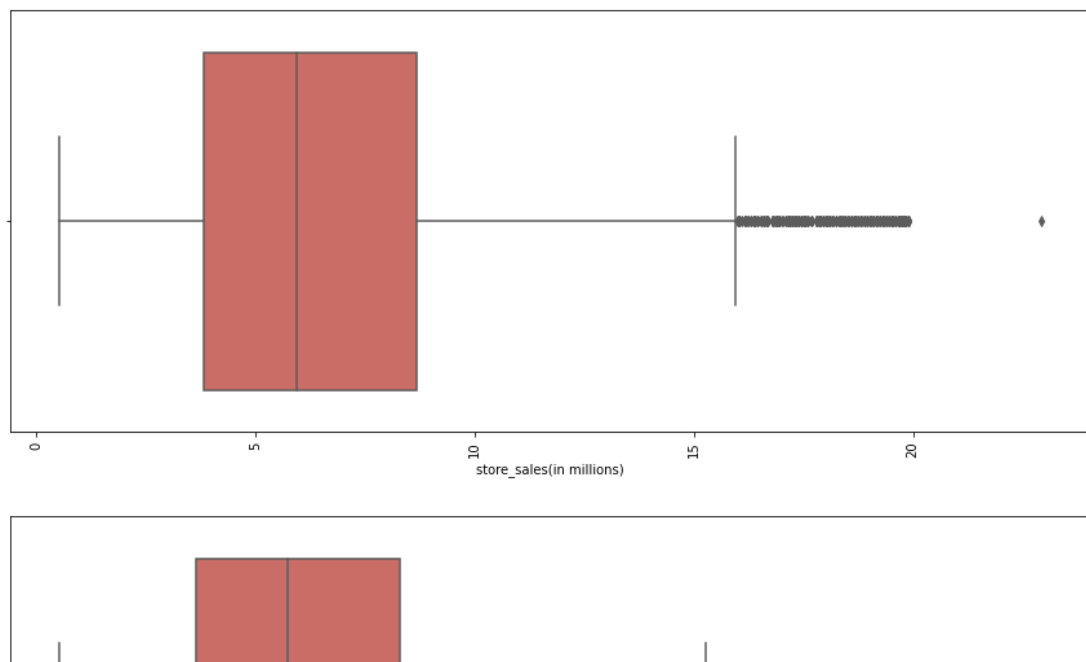
In [27]:

```
for i in num_cols:  
    plt.figure(figsize=(15,6))  
    sns.distplot(df[i], kde = True, bins = 20)  
    plt.xticks(rotation = 90)  
    plt.show()
```



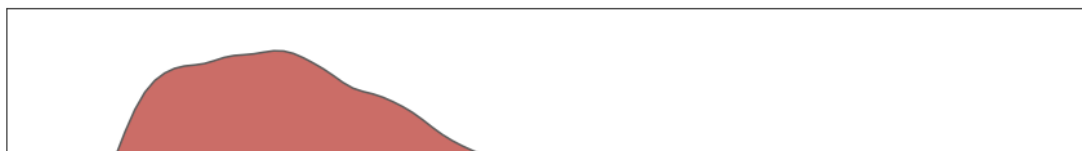
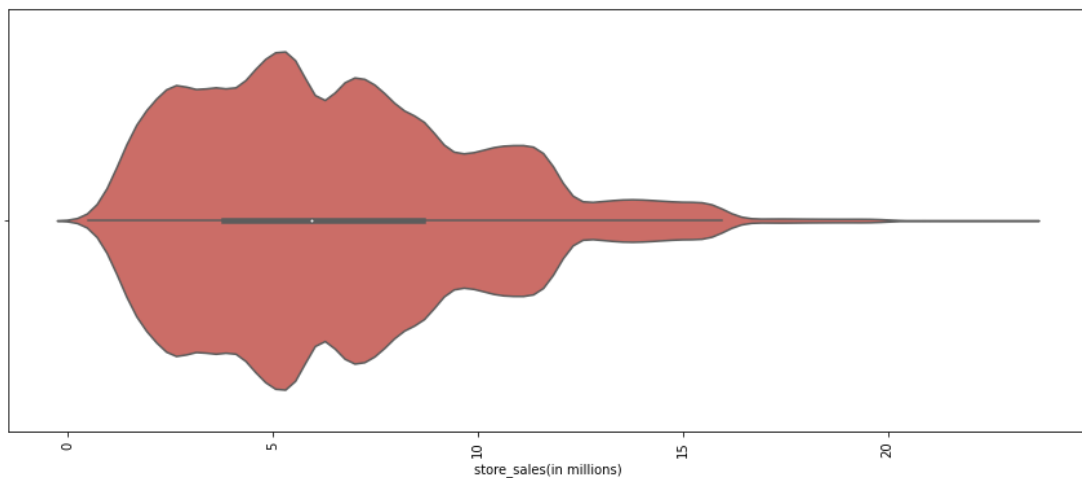
In [28]:

```
for i in num_cols:  
    plt.figure(figsize=(15,6))  
    sns.boxplot(df[i], data = df, palette = 'hls')  
    plt.xticks(rotation = 90)  
    plt.show()
```



In [29]:

```
for i in num_cols:  
    plt.figure(figsize=(15,6))  
    sns.violinplot(df[i], data = df, palette = 'hls')  
    plt.xticks(rotation = 90)  
    plt.show()
```



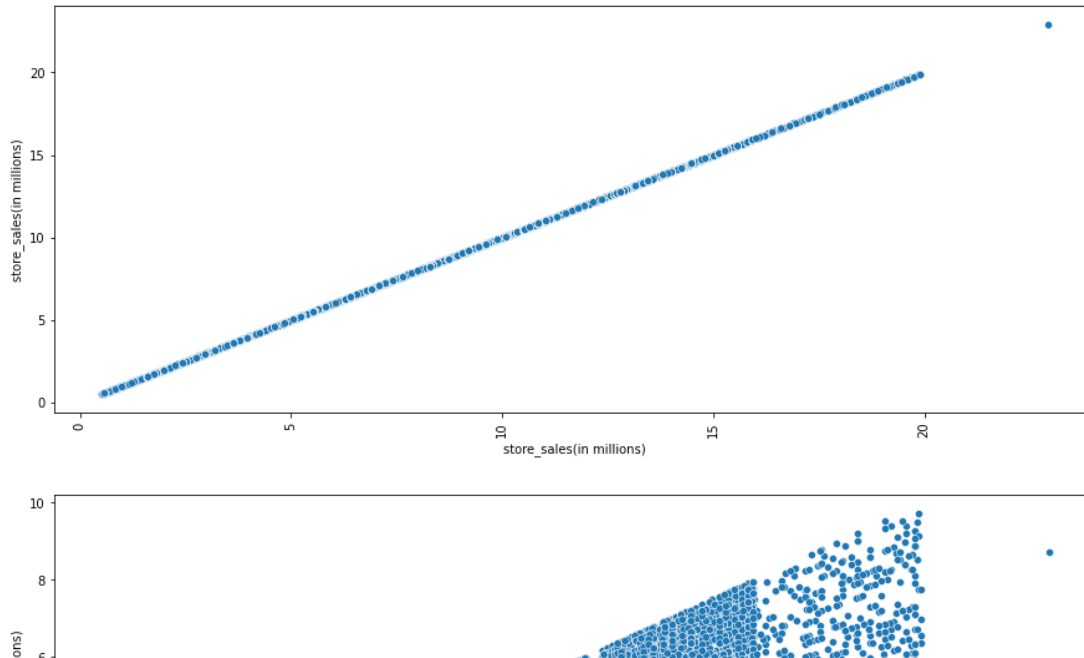
```
for i in num_cols: for j in num_cols: plt.figure(figsize=(15,6)) sns.lineplot(x = df[i], y = df[j], data = df, palette =  
'hls') plt.xticks(rotation = 90) plt.show()
```

In [31]:

```

for i in num_cols:
    for j in num_cols:
        plt.figure(figsize=(15,6))
        sns.scatterplot(x = df[i], y = df[j], data = df, palette = 'hls')
        plt.xticks(rotation = 90)
        plt.show()

```



```

for i in obj_cols: for j in num_cols: plt.figure(figsize=(15,6)) sns.barplot(x = df[i], y = df[j], data = df, palette =
'hls') plt.xticks(rotation = 90) plt.show()

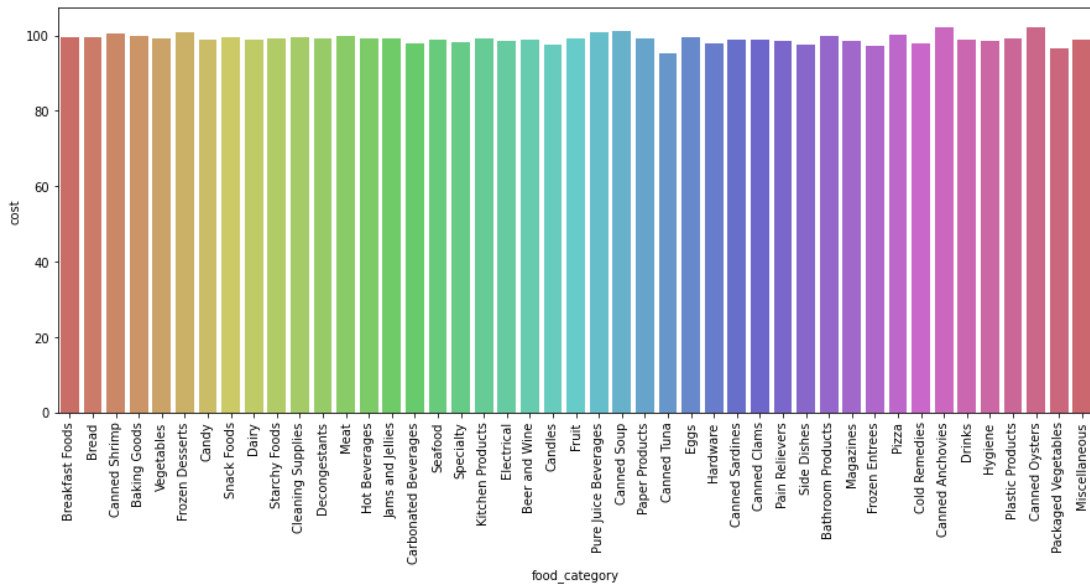
```

In [34]:

```

for i in obj_cols:
    plt.figure(figsize=(15,6))
    sns.barplot(x = df[i], y = df['cost'], data = df, ci = None, palette = 'hls')
    plt.xticks(rotation = 90)
    plt.show()

```



In [35]:

```
df_corr = df[num_cols].corr()
```

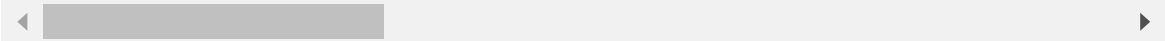
In [36]:

```
df_corr
```

Out[36]:

	store_sales(in millions)	store_cost(in millions)	unit_sales(in millions)	total_children	avg_cars_ home(appro
store_sales(in millions)	1.000000	0.954685	0.503482	0.083313	0.004498
store_cost(in millions)	0.954685	1.000000	0.480087	0.079058	0.002865
unit_sales(in millions)	0.503482	0.480087	1.000000	0.163188	0.023667
total_children	0.083313	0.079058	0.163188	1.000000	0.098110
avg_cars_at home(approx)	0.004498	0.002865	0.023667	0.098110	1.000000
num_children_at_home	0.032437	0.027576	0.066725	0.394709	0.130800
avg_cars_at home(approx).1	0.004498	0.002865	0.023667	0.098110	1.000000
SRP	0.833478	0.795880	-0.002358	0.000545	-0.007900
gross_weight	0.036179	0.034237	0.001255	-0.000186	0.004500
net_weight	0.032014	0.030257	0.001137	0.000142	0.004100
recyclable_package	0.034293	0.030213	0.001599	0.002794	0.003700
low_fat	-0.006134	-0.005976	-0.001129	-0.002824	-0.004300
units_per_case	-0.010630	-0.009792	0.000084	0.002307	-0.007200
store_sqft	0.015543	0.017877	0.031464	0.000555	-0.015800
grocery_sqft	0.010442	0.012884	0.024857	0.018526	-0.017600
frozen_sqft	0.017886	0.019245	0.030563	-0.026926	-0.007400
meat_sqft	0.017883	0.019242	0.030557	-0.026923	-0.007400
coffee_bar	-0.029368	-0.027126	-0.057633	0.002836	-0.002700
video_store	0.019179	0.019252	0.034996	-0.000591	0.014000
salad_bar	0.031459	0.033206	0.057878	-0.013764	-0.008900
prepared_food	0.031459	0.033206	0.057878	-0.013764	-0.008900
florist	0.030603	0.030929	0.055885	-0.003361	-0.004100
cost	-0.004621	-0.004162	-0.015015	-0.003900	0.011600

23 rows × 23 columns



In [38]:

```
plt.figure(figsize=(30, 10))
matrix = np.triu(df_corr)
sns.heatmap(df_corr, annot=True, linewidth=.8, mask=matrix, cmap="rocket");
plt.show()
```

